

INLIGTINGSTEGNOLOGIE: VRAESTEL II

Tyd: 3 uur

120 punte

LEES ASSEBLIEF DIE VOLGENDE INSTRUKSIES NOUKEURIG DEUR

1. Hierdie vraestel bestaan uit 14 bladsye. Maak asseblief seker dat jou vraestel volledig is.
2. Hierdie vraestel moet beantwoord word deur objekgeoriënteerde programmeringsbeginsels te gebruik. Jou program moet sinvol gebruik maak van metodes en parameters.
3. Hierdie vraestel is in twee afdelings verdeel. Alle kandidate moet albei afdelings beantwoord.
4. Hierdie vraestel is opgestel in programmeerterme wat nie eie is aan enige spesifieke programmeertaal (Java/Delphi) of databasis (Access/MySQL/JavaDB) nie.
5. Maak seker dat jy die vrae beantwoord op die wyse wat beskryf word, aangesien punte vir jou oplossing toegeken sal word ooreenkomstig die spesifikasies wat in die vraag gegee word.
6. Beantwoord slegs wat in elke vraag gevra word. As die vraag byvoorbeeld nie vra vir datageldigheidstoetsing nie, word geen punte daarvoor toegeken nie en hoef geen kode dus vir datageldigheidstoetsing geskryf te word nie.
7. As jy 'n gedeelte van die kode nie kan laat werk nie, moet jy dit as kommentaar merk sodat dit nie uitgevoer word nie en sodat jy met die eksamen kan voortgaan. Probeer indien moontlik om die fout te verduidelik om die nasiener te help.
8. Jou programme moet op so 'n manier gekodeer word dat hulle met enige data sal werk en nie slegs die voorbeelddata wat voorsien is of enige data-uittreksels wat in die vraestel voorkom nie. Jy word aangeraai om die datalêers wat voorsien word, noukeurig te bestudeer.

9. Maak seker dat roetines soos soektogte, sorterings en selekterings vir skikkings uit eerste beginsels ontwikkel word en dat jy nie die ingeboude eienskappe van 'n programmeertaal vir enigeen van hierdie roetines gebruik nie.
10. Alle datastrukture moet deur jou, die programmeerder, gedefinieer en verklaar word. Jy mag nie komponente wat binne die koppelvlak voorsien word, gebruik om data te stoor en later te herwin nie.
11. Lees die hele vraestel voordat jy 'n datastruktuur kies. Jy kan vind dat daar 'n alternatiewe metode is om die data voor te stel wat dalk meer doeltreffend sal wees met inagneming van die vrae wat in die vraestel gevra word.
12. Jy moet al jou werk gereeld stoor op die skyf wat jy ontvang het of die skyfspasie wat vir hierdie eksamen aan jou toegeken is. Jy moet voor jy begin ook 'n rugsteunkopie (*backup*) van die oorspronklike lêers skep ingeval die oorspronklike weergawe per ongeluk deur jou oplossing gewysig word.
13. Indien jou eksamen deur 'n tegniese probleem soos 'n kragonderbreking onderbreek word, sal jy, wanneer jy weer begin skryf, slegs die tyd gegee word wat oor was toe die onderbreking begin het om jou eksamen te voltooi. Geen ekstra tyd sal gegee word om werk wat nie gestoor is nie weer oor te doen nie.
14. Maak seker jou eksamennommer verskyn as kommentaar in elke program wat jy kodeer, sowel as op elke bladsy sigkopie (*hard copy*) wat jy inlewer.
15. Druk 'n kodelys van al die programme/klasse wat jy kodeer. Drukwerk moet na die eksamen gedoen word. Jy sal na afloop van die eksamen 'n halfuur kry om te druk. Jou onderwyser sal jou inlig oor die reëlins vir die druk van jou werk.
16. Jy moet van die volgende twee gidse (*folders*) (in vetdruk) en lêers (*files*) voorsien word. Hierdie lêers moet as data vir hierdie eksamen gebruik word. Let daarop dat die databasislêers in MS Access-, JavaDB- en MySQL-formaat voorsien word. Maak seker dat jy die lêers kan oopmaak met die pakkette wat jy sal gebruik om jou oplossings vir hierdie eksamen te kodeer.

Afdeling A:

HoopDB_Access.mdb
HoopDB_JavaDB.txt
HoopDB_MySQL.txt
SQLAntwoordblad.rtf
SQLBlaaier.exe

Afdeling B:

beraders.txt
kliente.txt

SCENARIO

'n Gemeenskapsgebaseerde beradingsdiens genaamd HOOP verskaf berading aan kliënte teen 'n verminderde bedrag genaamd 'n tarief. Kliënte kan veelvuldige afsprake by 'n berader maak by 'n bepaalde kantoor genaamd die plek.

AFDELING A SQL**VRAAG 1**

'n Databasis word gebruik om op hoogte te bly van kliënte en die afsprake wat hulle by beraders maak. Hierdie databasis bestaan uit drie tabelle:

KLIENT-tabel bevat die besonderhede van alle kliënte wat moontlik afsprake kan maak.

VELDE	DATATYPE	BESKRYWING
KlientID	HEELGETAL	'n Unieke outomatiese identifikasienommer vir elke kliënt
KlientNaam	TEKS	Die naam en van van die kliënt
Ouderdom	HEELGETAL	Die ouderdom van die kliënt wanneer hy/sy die eerste keer by HOOP aansluit
Poskode	TEKS	Die poskode van die kliënt se woonadres

BERADER-tabel bevat die besonderhede van alle beraders wat vir HOOP werk.

VELDE	DATATYPE	BESKRYWING
BeraderID	HEELGETAL	'n Unieke outomatiese identifikasienommer vir elke berader
BeraderNaam	TEKS	Die naam van die berader
Tarief	DUBBEL	Die tarief wat hierdie berader vra

AFSPRAAK-tabel bevat besonderhede van afsprake wat deur kliënte by 'n berader gemaak is vir 'n bepaalde datum en plek.

VELDE	DATATYPE	BESKRYWING
AfspraakID	HEELGETAL	'n Unieke outomatiese identifikasienommer vir elke afspraak
KlientID	HEELGETAL	Die KlientID van die kliënt wat hierdie afspraak gemaak het. Dit is 'n vreemde sleutel vir die Klient-tabel.
BeraderID	HEELGETAL	Die BeraderID van die berader by wie die afspraak gemaak is. Dit is 'n vreemde sleutel vir die Berader-tabel.
AfspraakDatum	DATUM	Die datum van die afspraak
Plek	TEKS	Die naam van die kantoor vir hierdie afspraak

- 1.1 Vertoon alle inligting oor kliënte wie se ouderdomme van 18 tot 30 jaar inklusief is. Orden hierdie lys in stygende volgorde volgens ouderdom. Die korrekte afvoer word hieronder getoon. Let daarop dat die sekondêre ordening kan verskil na gelang van die databasispakket wat gebruik word.

KlientID	KlientNaam	Ouderdom	Poskode
28	Cynthia Fourie	18	1500
8	Lindewe Khoza	18	1500
16	Maria Nkosi	21	0152
3	Steve Jacobs	21	0125
33	Bianca Abrahams	22	0160
26	Zandile Methembu	22	1600
2	Leo Sithole	22	1240
31	Al Naidoo	23	0127
23	Willem du Plessis	23	1600
4	Pat Khumalo	23	1251
30	Bongiwe Mokoena	25	0160
7	Sibongile Ngcobo	25	1240
37	Hendrick van Zyl	27	0180
15	Peter Zwane	30	1240

(3)

- 1.2 Vertoon die kliënte wat in die streek, waarvan die poskode met 012 begin, woon. Die korrekte afvoer word hieronder getoon. Let daarop dat die ordening kan verskil na gelang van die databasispakket wat gebruik word.

KlientID	KlientNaam	Ouderdom	Poskode
6	Musi Mahlangu	38	0120
18	Patricia Williams	17	0120
35	Moses Sibisi	31	0121
38	Lucky Shabangu	16	0123
3	Steve Jacobs	21	0125
31	Al Naidoo	23	0127

(3)

- 1.3 Kliëntkodes moet gegenereer word. Dit kan gedoen word deur die laaste twee letters van die kliëntnaam te neem en dit te kombineer met 'n ewekansige getal van 11 tot 17. Skryf 'n SQL-stelling om die kliëntkodes vir alle kliënte af te voer. Vertoon die nuwe kliëntkode langs die kliëntnaam. 'n Voorbeeld van die afvoer word hieronder getoon. Let daarop dat die kodes sal verskil na gelang van die ewekansige getal wat gegenereer word.

Let daarop dat slegs die eerste vyf rekords getoon word.

KlientNaam	Kode <i>Let daarop dat die laaste twee syfers lukraak gegenereer word</i>
John Dlamini	ni12
Leo Sithole	le15
Steve Jacobs	bs12
Pat Khumalo	lo17
Sipho Nkosi	si15
...	...

(5)

- 1.4 Vertoon die KlientID van kliënte wat nog nooit 'n afspraak in Bergsig, Panorama of Highlands gehad het nie. 'n Voorbeeld van die korrekte afvoer word hieronder getoon.

Let daarop dat slegs die eerste vyf rekords getoon word.

KlientID	Plek
2	Greenside
5	Greenside
24	Greenside
29	Greenside
36	Greenside
...	...

(3)

- 1.5 Nuwe beraders het 'n tarief van nul terwyl hulle wag dat hul tarief toegeken word. Vertoon die beradername met die laagste nienultarief. Die korrekte afvoer word hieronder getoon.

BeraderNaam	Tarief
Siyanda Mabuza	120
Joshua Hendricks	120
Linda September	120

(5)

- 1.6 Vertoon die plekke waar meer as 15 toekomstige afspraak gemaak is. Vertoon elke plek en die getal afspraak. Moenie afspraak in die verlede insluit nie. Die korrekte afvoer word hieronder getoon.

Let daarop dat die afvoer kan verander na gelang van vandag se datum of die datuminstelling van jou rekenaar.

Plek	GetalAfsprake
Bergsig	22
Panorama	17

(6)

- 1.7 Alle kliënte jonger as 15 ontvang 25% afslag. Vertoon die nuwe tarief van alle afspraak wat kliënte jonger as 15 behels. Vertoon afspraak-ID, kliënt-ID, kliëntnaam, beradernaam en afslagkoste. 'n Voorbeeld van die korrekte afvoer word hieronder getoon.

Let daarop dat wanneer die lys volgens KlientID georden word, hierdie vyf korrekte rekords getoon sal word. Let ook daarop dat data anders geformateer kan wees op jou rekenaar.

Afspraak ID	Klient ID	KlientNaam	BeraderNaam	AfslagKoste
6	5	Sipho Nkosi	Joshua Hendricks	90
35	14	Mpho Sithole	Thabo Matlala	112.5
69	14	Mpho Sithole	Thabo Matlala	112.5
24	25	Linda Gumede	Matthew Kunene	135
44	25	Linda Gumede	Matthew Kunene	135
...

(5)

- 1.8 Die Bergsig-kantoor het op 24 Oktober 2020 vir opknapping gesluit. Alle afspraak by hierdie plek sal verander moet word. Dit sal na die Middelburg-kantoor verskuif word. Skryf 'n SQL-stelling om dit te doen.
- 1.9 Voeg 'n nuwe opvolgafspraak vir KlientID 27 in vir 20 Oktober 2020. Kopieer ander inligting (BeraderID en Plek) uit die kliënt se laaste afspraak (dit is die een met AfspraakID 18).

(4)

(6)

40 punte

AFDELING B OBJEKGEORIËNTEERDE PROGRAMMERING

'n Gemeenskapsgebaseerde beradingsdiens genaamd HOOP maak dit vir kliënte moontlik om 'n tydgleuf te bespreek om hul berader te sien. Elke tydgleuf is een uur lank. Die kliënte se besonderhede word in 'n tekslêer genaamd **kliente.txt** gestoor.

'n Besprekingskedsule moet op 'n bepaalde dag vir vyf van die beraders geskep word. Die name van die beraders word gelys in 'n tekslêer genaamd **beraders.txt**. Vir al die kliënte moet 'n afspraak by hul berader bespreek word op 'n tyd later as of gelyk aan hul voorkeurtyd.

Die program vereis die volgende basisklasse:

Klient

'n Kliënt kan 'n tydgleuf by enige berader bespreek en het die volgende inligting:

- **klientNaam** – die naam van die kliënt
- **voorkeurBerader** – die naam van die berader wat die kliënt sal verkies om te sien.
- **vroegsteUur** – die vroegste tydgleuf wat die kliënt sal wil bespreek. 8 (wat 8:00 verteenwoordig) is die vroegste, terwyl 16 (wat 16:00 verteenwoordig) die laatste gleuf is. LET WEL: 24-uur-notasie word gebruik met waardes van 0 tot 23.

TydGleuf

'n Klas moet geskep word om elke tydgleuf by 'n berader te verteenwoordig. Elke tydgleuf is een uur lank. 'n **TydGleuf** het die volgende inligting:

- **berader** – die naam van die berader vir die tydgleuf
- **beginUur** – die beginuur van die tydgleuf in 24-uur-tydformaat. Vir 'n TydGleuf wat om 9 begin, sal hierdie veld byvoorbeeld op 9 gestel wees, terwyl 'n tydgleuf wat om 14:00 begin, 14 in hierdie veld sal hê.
- **isBeskikbaar** – of hierdie tydgleuf beskikbaar is vir bespreking.

Die tekslêer genaamd **kliente.txt** stoor die besonderhede van die tydgleuwe wat kliënte verlang. Die kliënt se naam, berader en die vroegste tydgleuf word elkeen in 'n afsonderlike reël gelys. 'n Voorbeeld van die eerste vyf voorkeurbesprekings word hieronder gelys:

```
John Dlamini
Vernon Booysen
10
Leo Sithole
Matthew Kunene
13
Steve Jacobs
Vernon Booysen
8
Pat Khumalo
Heather Modise
6
Sipho Nkosi
Vernon Booysen
18
```

VRAAG 2

Gebruik die klasdiagram hieronder om 'n nuwe klas genaamd **Klient** te skep. Hierdie klas sal gebruik word om die besonderhede van 'n kliënt te stoor. Die diagram hieronder dui die velde en metodes aan wat vereis word.

Klient
- klientNaam : string - voorkeurBerader : string - vroegsteUur : integer
+ Konstruktor(inKN : string, inVBR : string, inVU : integer) + kryKlientNaam() : string + kryVoorkeurBerader() : string + kryVroegsteUur() : integer + toString() : string

2.1 Skep 'n nuwe klas genaamd **Klient** met **klientNaam**-, **voorkeurBerader**- en **vroegsteUur**-veld soos hierbo aangedui. (4)

2.2 Skep 'n konstruktormetode wat 'n string **inKN** as parameter aanvaar wat die **klientNaam**-veld verteenwoordig, 'n string **inVBR** as parameter wat die **voorkeurBerader**-veld verteenwoordig en 'n heelgetal **inVU** wat die **vroegsteUur** verteenwoordig. Gebruik hierdie parameters om waardes aan die velde toe te ken. Let daarop dat **vroegsteUur** op 16 gestel moet word vir 'n waarde in **inVU** groter as 16 om te verseker dat die gleuwe binne die beskikbare tyd val. Daar is geen probleem met getalle kleiner as 8 nie, aangesien die vroegste beskikbare gleuf 8 is. (4)

2.3 Skep toegangser/kry-metodes vir die **klientNaam**, **voorkeurBerader** en **vroegsteUur**-velde. (2)

2.4 Kodeer 'n **toString()**-metode om die velde van 'n **Klient** as 'n string terug te stuur in die volgende formaat:

```
<klientNaam><tab><voorkeurBerader><tab>Vroegste <vroegsteUur>:00
```

Byvoorbeeld:

```
John Dlamini      Vernon Booyen      Vroegste 10:00
```

(4)
[14]

VRAAG 3

Skep 'n **TydGleuf**-klas wat deur die volgende klasdiagram verteenwoordig word:

TydGleuf
- berader : string - beginUur : integer - isBeskikbaar : boolean
+ Konstruktor(inBR : string, inBU : integer) + stelIsBeskikbaar(inIB : boolean) + kryBeraderNaam() : string + kryBeginUur() : integer + kryIsBeskikbaar() : boolean - kryEindUur() : integer + toString() : string

- 3.1 Skep 'n nuwe klas genaamd **TydGleuf** met **berader**-, **beginUur**- en **isBeskikbaar**-veld soos hierbo aangedui. (3)
- 3.2 Skep 'n konstruktormetode wat 'n string **inBR** as parameter aanvaar wat die **berader**-veld verteenwoordig en 'n heelgetal **inBU** wat die **beginUur**-veld verteenwoordig. Gebruik hierdie parameters om waardes aan die velde toe te ken. Die veld **isBeskikbaar** moet by verstek op *true* gestel word. (4)
- 3.3 Skep 'n mutator/stel-metode vir die **isBeskikbaar**-veld. (2)
- 3.4 Skep toegangster/kry-metodes vir die **beraderNaam**-, **beginUur**- en **isBeskikbaar**-veld. (1)
- 3.5 Kodeer 'n privaat metode genaamd **kryEindUur** wat die uur terugstuur waarop die **TydGleuf** eindig. Onthou dat elke **TydGleuf** een uur lank is. 'n **Tydgleuf** wat byvoorbeeld om 13 begin, sal om 14 eindig. (2)
- 3.6 Kodeer 'n **toString()**-metode om inligting oor 'n **TydGleuf** terug te stuur in die formaat:

<BeraderNaam>: <BeginUur>:00 - <EindUur>:00

Byvoorbeeld:

John Dlamini: 10:00 - 11:00

Let daarop dat die metode wat in 3.5 geskep is, gebruik moet word om volpunte vir hierdie vraag te verdien.

(4)
[16]

VRAAG 4

'n Nuwe klas **GleufBestuurder** moet geskep word.

- 4.1 Skep 'n nuwe klas genaamd **GleufBestuurder** met twee privaat velde soos volg:
- 'n Skikking van **Klient**-objekte genaamd **kSkik** met genoeg kapasiteit om 20 **Klient**-objekte te stoor
 - 'n Skikking van **TydGleuf**-objekte genaamd **tSkik** met genoeg kapasiteit om 40 **TydGleuf**-objekte te stoor
- (4)
- 4.2 Skryf kode vir die konstruktormetode van die **GleufBestuurder**-klas wat geen parameter inneem nie. Die konstruktormetode sal die inhoud van 'n tekslêer genaamd **kliente.txt** lees. Drie reëls word gebruik om inligting oor elke kliënt te stoor – kliëntnaam, voorkeurberader en voorkeurtydgleuf in hierdie volgorde. Daar is presies 60 reëls in die lêer.
- Doen die volgende:
- Maak die lêer oop om te lees.
 - Loop in 'n lus deur die lêer om 60 reëls te lees. Skep vir elke drie reëls wat uit die tekslêer gelees word 'n **Klient**-objek en voeg die objek by die skikking genaamd **kSkik**.
- (8)
- 4.3 Skep 'n getipeerde metode/funksie genaamd **vertoonAlleKliente()** in die **GleufBestuurder**-klas wat 'n string sal terugstuur wat al die **Klient**-objekte in die **Klient**-skikking genaamd **kSkik** bevat. Die **Klient**-objekte moet ingesluit word in die formaat van die toString-metode van die **Klient**-klas met elke kliënt se inligting op 'n afsonderlike reël.
- (4)
- [16]**

VRAAG 5

- 5.1 Skryf kode om 'n teksgebaseerde gebruikerskoppelvlak genaamd **BeradingGK** te skep wat eenvoudige afvoer moontlik sal maak. (1)
- 5.2 Skep 'n **GleufBestuurder**-objek deur die toepaslike metode te gebruik. (1)
- 5.3 Skryf kode om al die kliënte se inligting te vertoon.

'n Voorbeeld van die afvoer word hieronder getoon:

John Dlamini	Vernon Booysen	Vroegste 10:00
Leo Sithole	Matthew Kunene	Vroegste 13:00
Steve Jacobs	Vernon Booysen	Vroegste 8:00
Pat Khumalo	Heather Modise	Vroegste 6:00
Sipho Nkosi	Vernon Booysen	Vroegste 16:00

(1)
[3]

VRAAG 6

'n Besprekingslys moet gegenereer word vir party van die beraders wat in 'n tekslêer genaamd **beraders.txt** gelys word. Elke berader benodig 'n lys van ure wat om 8:00 begin en om 16:00 eindig met die uitsondering van 12:00 wat vir 'n etensuur gereserveer word.

Die name word gestoor op 'n enkele reël wat deur kommas geskei word soos hieronder getoon:

Vernon Booysen,Matthew Kunene,Heather Modise,Siyanda Mabuza,Joshua Hendricks
--

6.1 Skep in die **GleufBestuurder**-klas 'n leë metode/prosedure genaamd **genereerTydGleuwe()** wat uit die tekslêer **beraders.txt** lees.

Skryf kode om een reël van die tekslêer te lees, elke berader se naam te isoleer en die volgende te doen:

- Elke berader sal agt tydgleuwe benodig wat met sy/haar naam as die berader geskep word. Die eerste **TydGleuf** vir elke konsultant sal 08:00 begin en die laaste **TydGleuf** sal 16:00 begin. Daar is elke uur 'n gleuf, behalwe vir die tydgleuf wat 12:00 begin, aangesien die beraders 'n etensuur het.
- Daar sal agt **TydGleuf**-objekte vir elkeen van die vyf beraders geskep word, wat 40 **TydGleuf**-objekte tot gevolg sal hê. Voeg al 40 **TydGleuf**-objekte in die **TydGleuf**-skikking genaamd **tSkik** in. Die objekte kan in enige volgorde ingevoeg word.

Punte word toegeken vir die doeltreffende gebruik van lusse om verwerkingstyd te verminder.

(11)

6.2 Skep in die **GleufBestuurder**-klas 'n getipeerde metode/funksie genaamd **vertoonBeskikbareTydGleuwe(..)** wat 'n string sal terugstuur wat al die beskikbare tydgleuwe vir elkeen van die vyf beraders in die **TydGleuf**-skikking genaamd **tSkik** bevat. Die **TydGleuf**-objekte moet ingesluit word in die formaat van die toString-metode van die **TydGleuf**-klas met elke tydgleuf se inligting op 'n afsonderlike reël.

(4)

6.3 Voeg kode in die **BeradingGK**-klas by om die volgende te doen:

- Roep **genereerTydGleuwe()** in die **GleufBestuurder**-klas wat in Vraag 6.1 geskep is.
- Roep **vertoonBeskikbareTydGleuwe()** in die **GleufBestuurder**-klas in Vraag 6.2.

Moontlike afvoer word hieronder getoon. Let daarop dat die lys in enige volgorde kan wees:

```
Vernon Booysen: 8:00 - 9:00
Vernon Booysen: 9:00 - 10:00
Vernon Booysen: 10:00 - 11:00
Vernon Booysen: 11:00 - 12:00
Vernon Booysen: 13:00 - 14:00
Vernon Booysen: 14:00 - 15:00
Vernon Booysen: 15:00 - 16:00
Vernon Booysen: 16:00 - 17:00
Matthew Kunene: 8:00 - 9:00
Matthew Kunene: 9:00 - 10:00
Matthew Kunene: 10:00 - 11:00
Matthew Kunene: 11:00 - 12:00
Matthew Kunene: 13:00 - 14:00
Matthew Kunene: 14:00 - 15:00
Matthew Kunene: 15:00 - 16:00
Matthew Kunene: 16:00 - 17:00
Heather Modise: 8:00 - 9:00
Heather Modise: 9:00 - 10:00
Heather Modise: 10:00 - 11:00
Heather Modise: 11:00 - 12:00
Heather Modise: 13:00 - 14:00
Heather Modise: 14:00 - 15:00
Heather Modise: 15:00 - 16:00
Heather Modise: 16:00 - 17:00
Siyanda Mabuza: 8:00 - 9:00
Siyanda Mabuza: 9:00 - 10:00
Siyanda Mabuza: 10:00 - 11:00
Siyanda Mabuza: 11:00 - 12:00
Siyanda Mabuza: 13:00 - 14:00
Siyanda Mabuza: 14:00 - 15:00
Siyanda Mabuza: 15:00 - 16:00
Siyanda Mabuza: 16:00 - 17:00
Joshua Hendricks: 8:00 - 9:00
Joshua Hendricks: 9:00 - 10:00
Joshua Hendricks: 10:00 - 11:00
Joshua Hendricks: 11:00 - 12:00
Joshua Hendricks: 13:00 - 14:00
Joshua Hendricks: 14:00 - 15:00
Joshua Hendricks: 15:00 - 16:00
Joshua Hendricks: 16:00 - 17:00
```

VRAAG 7

- 7.1 Kodeer 'n metode genaamd **skepBespreekteGleufSkikking()** in die **GleufBestuurder**-klas. Die metode sal vir elke kliënt 'n tydgleuf kies by die **Klient** se voorkeurberader (soos deur die **Klient** se **voorkeurBeraderNaam**-veld aangedui) en die vroegste uur wat die kliënt kan kom (soos deur die **Klient** se **vroegsteUur**-veld aangedui).

Let op die volgende:

- Slegs een kliënt kan in 'n berader se tydgleuf bespreek word.
- Daar is genoeg tydgleuwe in die **TydGleuf**-skikking om ooreenkomstig al die **Klient**-voorkeure toegeken te word.
- Indien die kliënt se voorkeurtijd voor 8:00 is, moet die kliënt 'n tydgleuf ontvang wat van 8:00 af begin.

Die metode moet 'n string terugstuur met die volgende formaat:

Afsprake:

```
<klientNaam1> (<vroegsteUur>) sien <beraderNaam>: <BU>:00 - <EU>:00
<klientNaam2> (<vroegsteUur>) sien <beraderNaam>: <BU>:00 - <EU>:00
<klientNaam3> (<vroegsteUur>) sien <beraderNaam>: <BU>:00 - <EU>:00
...
```

waar <BU> en <EU> onderskeidelik <beginUur> en <eindUur> is.

Hieronder is voorbeeldafvoer met slegs die eerste vyf kliënte se toekennings wat getoon word.

Afsprake:

```
John Dlamini (10) sien Vernon Booysen: 10:00 - 11:00
Leo Sithole (13) sien Matthew Kunene: 13:00 - 14:00
Steve Jacobs (8) sien Vernon Booysen: 8:00 - 9:00
Pat Khumalo (6) sien Heather Modise: 8:00 - 9:00
Sipho Nkosi (16) sien Vernon Booysen: 16:00 - 17:00
...
```

Let daarop dat jou afvoer kan verskil wat betref volgorde en tyd wat aan 'n kliënt toegeken word solank 'n kliënt aan die voorkeurberader toegeken word en 'n tydgleuf het wat nie vroeër is as die **vroegsteUur**-veld van die **Klient** nie.

(13)

- 7.2 Voeg kode by **BeradingGK** om die afspraak te roep en te vertoon deur die **skepBespreekteGleufSkikking()**-metode wat jy in Vraag 7.1 geskep het, te gebruik.

(1)
[14]

80 punte

Totaal: 120 punte